Name: Carlo Finnegan
Student No: 18379666

Exercise 1:

I created the basic microservices infrastructure setting up both UserService and BookService. I created a Dockerfile for each service, configured a PostgreSQL database, and established the necessary Docker Compose configuration. The UserService runs on port 5002 and handles user management operations , the BookService operates on port 5006 managing book-related operations. I tested all endpoints using curl commands to verify the functionality of both services as seen below.

#run docker compose
sudo docker compose up

User Service Testing

Create users
curl -X POST http://localhost:5002/users/add -H "Content-Type: application/json" -d '{"studentid": "1", "firstname": "Carlo", "lastname": "Finnegan", "email": "carlo.finnegan@ucdconnect.ie"}'
curl -X POST http://localhost:5002/users/add -H "Content-Type: application/json" -d '{"studentid": "2", "firstname": "Carlo2", "lastname": "Finnegan2", "email": "carlo2.finnegan@ucdconnect.ie"}'

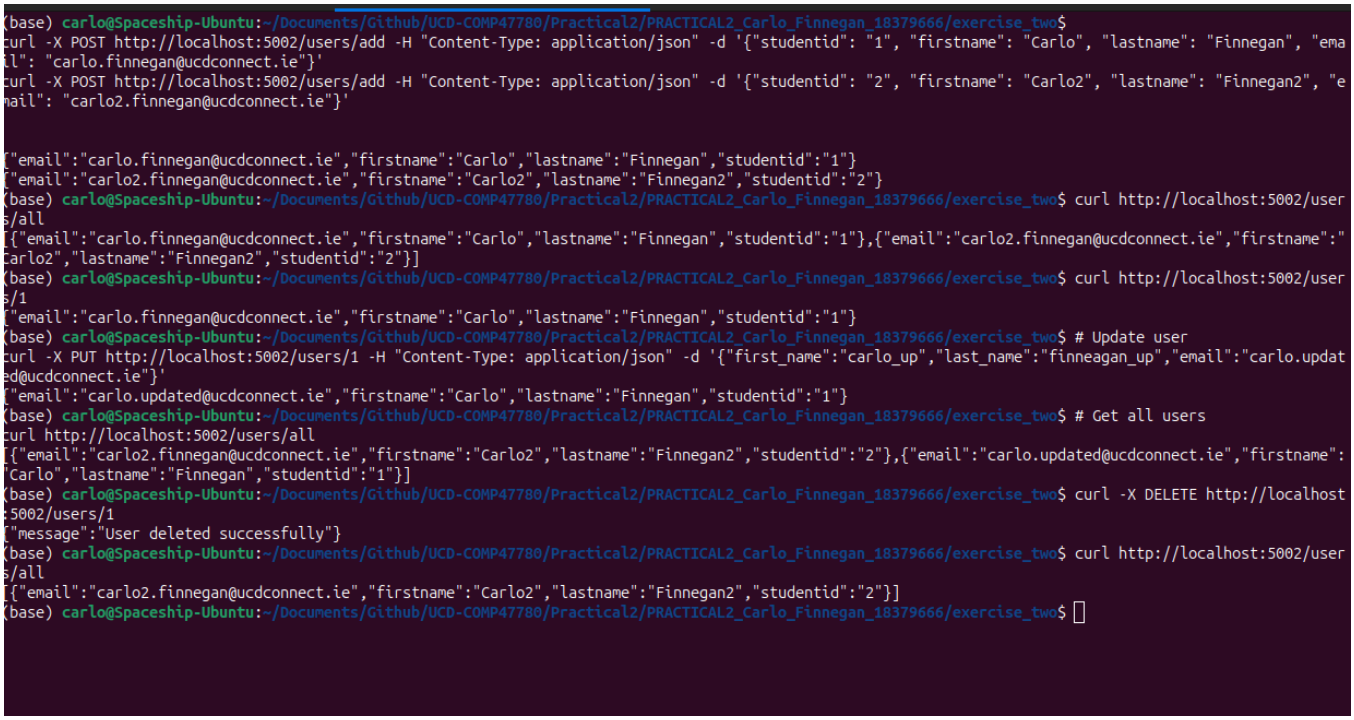
Get all users
curl http://localhost:5002/users/all

Get specific user
curl http://localhost:5002/users/1

Update user
curl -X PUT http://localhost:5002/users/1 -H "Content-Type: application/json" -d '{"first_name":"carlo_up","last_name":"finneagan_up","email":"carlo.updated@ucdconnect.ie"}'

Delete user
curl -X DELETE http://localhost:5002/users/2

Name: Carlo Finnegan
Student No: 18379666



Book Service Testing

I then setup my book service following instructions:
Adding new books (book_id, title, author) and any other information necessary.

Create books
**curl -X POST -H "Content-Type: application/json" -d '{"bookid":"1","title":"book1","author":"carlo1"}' http://localhost:5006/books/add; **
**curl -X POST -H "Content-Type: application/json" -d '{"bookid":"2","title":"book2","author":"carlo2"}' http://localhost:5006/books/add; **
**curl -X POST -H "Content-Type: application/json" -d '{"bookid":"3","title":"book3","author":"carlo3"}' http://localhost:5006/books/add; **
**curl -X POST -H "Content-Type: application/json" -d '{"bookid":"4","title":"book4","author":"carlo4"}' http://localhost:5006/books/add; **
**curl -X POST -H "Content-Type: application/json" -d '{"bookid":"5","title":"book5","author":"carlo5"}' http://localhost:5006/books/add; **
**curl -X POST -H "Content-Type: application/json" -d '{"bookid":"6","title":"book6","author":"carlo6"}' http://localhost:5006/books/add; **

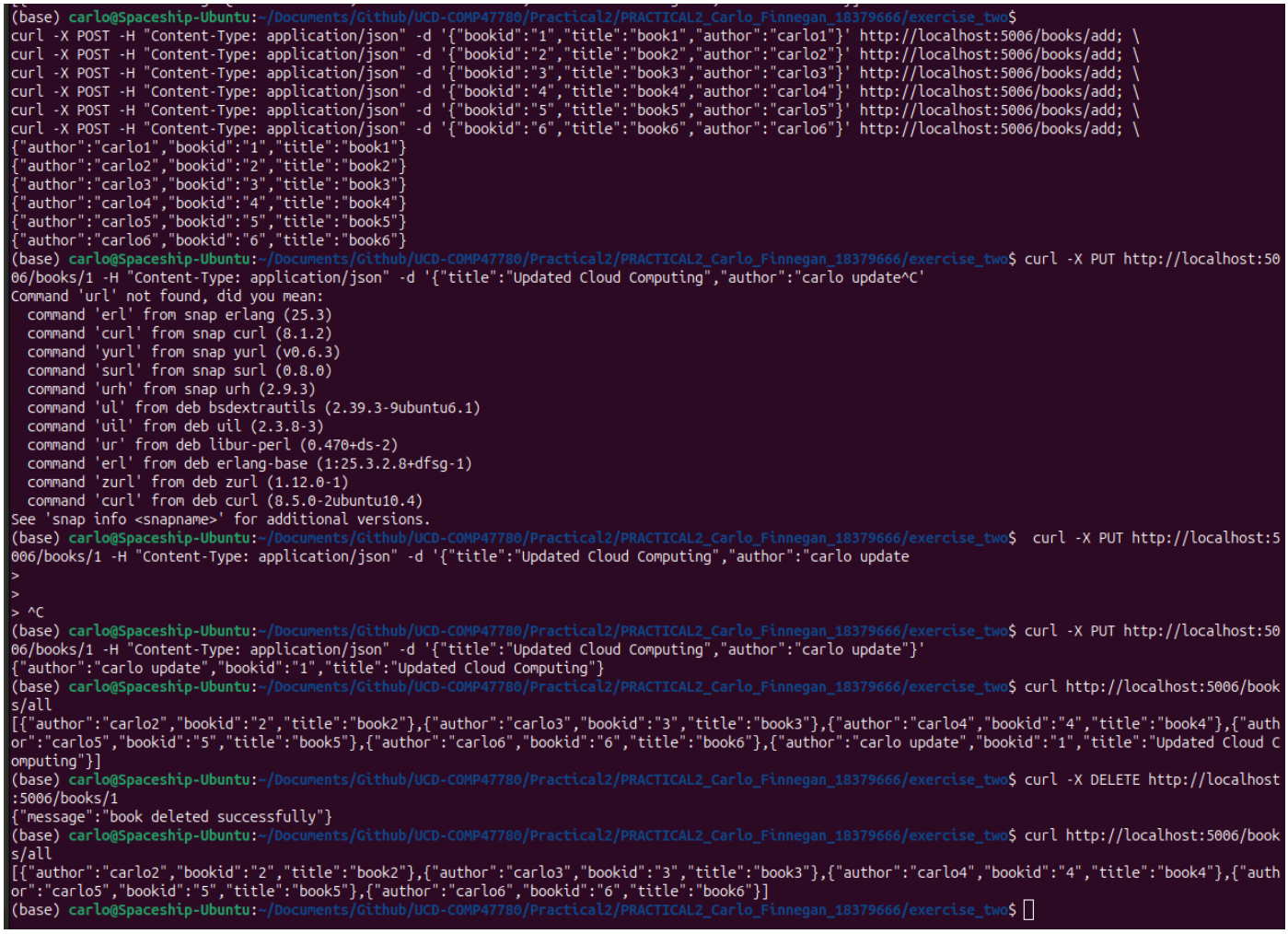
Get all books
curl http://localhost:5006/books/all

Get specific books for student id
curl http://localhost:5006/books/1

Name: Carlo Finnegan
Student No: 18379666

Update book
curl -X PUT http://localhost:5006/books/1 -H "Content-Type: application/json" -d '{"title":"Updated Cloud Computing","author":"carlo update"}'

Delete book
curl -X DELETE http://localhost:5006/books/1



Exercise 2

I implemented asynchronous communication between services using RabbitMQ. I modified the UserService to publish borrow requests to a RabbitMQ channel and developed the BorrowService to consume these messages. I setup a 5 book student limit and maintains borrow records in the database. The services communicate using the apis. New functionality was tested using curl commands. I also added logging to verify when borrow requests were invalid such as exceeding book limit.

Name: Carlo Finnegan
Student No: 18379666

To start my service
sudo docker compose up

#Do 5 borrow requests
**curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"1","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/request; **
**curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"2","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/request; **
**curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"3","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/request; **
**curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"4","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/request; **
**curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"5","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/request; **

#Borrow Book 6 for User 1 (rejected)

curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"6","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/request;

Retrieve Borrowed Books for User 1
curl -X GET http://localhost:5008/borrows/1;

Name: Carlo Finnegan
Student No: 18379666

```
(base) carlo@Spaceship-Ubuntu:~/Documents/Github/UCD-COMP47780/Practical2/PRACTICAL2_Carlo_Finnegan_18379666/exercise_two$
curl -X POST http://localhost:5002/users/add -H "Content-Type: application/json" -d '{"studentid": "1", "firstname": "Carlo", "lastname": "Finnegan", "ema
il": "carlo.finnegan@ucdconnect.ie"}'
curl -X POST http://localhost:5002/users/add -H "Content-Type: application/json" -d '{"studentid": "2", "firstname": "Carlo2", "lastname": "Finnegan2", "e
mail": "carlo2.finnegan@ucdconnect.ie"}'

curl http://localhost:5002/users/all



curl -X POST -H "Content-Type: application/json" -d '{"bookid":"1","title":"book1","author":"carlo1"}' http://localhost:5006/books/add; \
curl -X POST -H "Content-Type: application/json" -d '{"bookid":"2","title":"book2","author":"carlo2"}' http://localhost:5006/books/add; \
curl -X POST -H "Content-Type: application/json" -d '{"bookid":"3","title":"book3","author":"carlo3"}' http://localhost:5006/books/add; \
curl -X POST -H "Content-Type: application/json" -d '{"bookid":"4","title":"book4","author":"carlo4"}' http://localhost:5006/books/add; \
curl -X POST -H "Content-Type: application/json" -d '{"bookid":"5","title":"book5","author":"carlo5"}' http://localhost:5006/books/add; \
curl -X POST -H "Content-Type: application/json" -d '{"bookid":"6","title":"book6","author":"carlo6"}' http://localhost:5006/books/add; \


curl http://localhost:5006/books/all
{"email":"carlo.finnegan@ucdconnect.ie","firstname":"Carlo","lastname":"Finnegan","studentid":"1"}
{"email":"carlo2.finnegan@ucdconnect.ie","firstname":"Carlo2","lastname":"Finnegan2","studentid":"2"}
[{"email":"carlo.finnegan@ucdconnect.ie","firstname":"Carlo","lastname":"Finnegan","studentid":"1"},{"email":"carlo2.finnegan@ucdconnect.ie","firstname":"
Carlo2","lastname":"Finnegan2","studentid":"2"}]
{"author":"carlo1","bookid":"1","title":"book1"}
{"author":"carlo2","bookid":"2","title":"book2"}
{"author":"carlo3","bookid":"3","title":"book3"}
{"author":"carlo4","bookid":"4","title":"book4"}
{"author":"carlo5","bookid":"5","title":"book5"}
{"author":"carlo6","bookid":"6","title":"book6"}
[{"author":"carlo1","bookid":"1","title":"book1"},{"author":"carlo2","bookid":"2","title":"book2"},{"author":"carlo3","bookid":"3","title":"book3"},{"auth
or":"carlo4","bookid":"4","title":"book4"},{"author":"carlo5","bookid":"5","title":"book5"},{"author":"carlo6","bookid":"6","title":"book6"}]
(base) carlo@Spaceship-Ubuntu:~/Documents/Github/UCD-COMP47780/Practical2/PRACTICAL2_Carlo_Finnegan_18379666/exercise_two$
curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"1","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/reque
st; \
curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"2","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/reque
st; \
curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"3","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/reque
st; \
curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"4","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/reque
st; \
curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"5","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/reque
st; \

# Attempt to Borrow Book 6 for User 1 (should be rejected)
curl -X POST -H "Content-Type: application/json" -d '{"studentid":"1","bookid":"6","date_returned":"2023-12-31"}' http://localhost:5002/users/borrow/reque
st; \
{"message":"Borrow request succesfully posted","request":{"bookid":"1","date_returned":"2023-12-31","studentid":"1"}}
{"message":"Borrow request succesfully posted","request":{"bookid":"2","date_returned":"2023-12-31","studentid":"1"}}
{"message":"Borrow request succesfully posted","request":{"bookid":"3","date_returned":"2023-12-31","studentid":"1"}}
{"message":"Borrow request succesfully posted","request":{"bookid":"4","date_returned":"2023-12-31","studentid":"1"}}
{"message":"Borrow request succesfully posted","request":{"bookid":"5","date_returned":"2023-12-31","studentid":"1"}}
>
{"message":"Borrow request succesfully posted","request":{"bookid":"6","date_returned":"2023-12-31","studentid":"1"}}
(base) carlo@Spaceship-Ubuntu:~/Documents/Github/UCD-COMP47780/Practical2/PRACTICAL2_Carlo_Finnegan_18379666/exercise_two$ curl -X GET http://localhost:50
08/borrows/1; \
>
[{"bookid":"1","date_borrowed":"2024-11-15","date_returned":"2023-12-31","id":1,"studentid":"1"},{"bookid":"2","date_borrowed":"2024-11-15","date_returned
":"2023-12-31","id":2,"studentid":"1"},{"bookid":"3","date_borrowed":"2024-11-15","date_returned":"2023-12-31","id":3,"studentid":"1"},{"bookid":"4","date
_borrowed":"2024-11-15","date_returned":"2023-12-31","id":4,"studentid":"1"},{"bookid":"5","date_borrowed":"2024-11-15","date_returned":"2023-12-31","id":
5,"studentid":"1"}]
(base) carlo@Spaceship-Ubuntu:~/Documents/Github/UCD-COMP47780/Practical2/PRACTICAL2_Carlo_Finnegan_18379666/exercise_two$
```

Borrow rejection can be seen in logs and it can be seen no book 6 is present in the book list for that student. Can also see the get queries being used by borrow request

```
user    | 172.18.0.1 - - [15/Nov/2024 23:39:01] "POST /users/borrow/request HTTP/1.1" 201 -
borrow  | 2024-11-15 23:39:01,083 INFO Received borrow request: {'studentid': '1', 'bookid': '6', 'date_returned': '2023-12-31'}
user    | 172.18.0.5 - - [15/Nov/2024 23:39:01] "GET /users/1 HTTP/1.1" 200 -
book    | 172.18.0.5 - - [15/Nov/2024 23:39:01] "GET /books/6 HTTP/1.1" 200 -
borrow  | 2024-11-15 23:39:01,090 INFO Student 1 has reached borrow limit
borrow  | 2024-11-15 23:39:15,123 INFO 172.18.0.1 - - [15/Nov/2024 23:39:15] "GET /borrows/1 HTTP/1.1" 200 -
```

Name: Carlo Finnegan
Student No: 18379666

Exercise 3

I migrated the application to Kubernetes using Kompose to convert the Docker Compose configuration into Kubernetes manifests. I deployed the services to my Kubernetes cluster and established port forwarding to access each service (UserService:5002, BookService:5006, BorrowService:5008, Database:5432, RabbitMQ:5672/15672). I published all service images to my Docker Hubregistry to allow clean deployment

start in exercise 3 directory
cd UserService
sudo docker build -t carlofinnegan/userservice:latest .

cd ../BorrowService
sudo docker build -t carlofinnegan/borrowservice:latest .

cd ../BookService
sudo docker build -t carlofinnegan/bookservice:latest .

cd ..
#Ensure i login to dockerhub in cli then
sudo docker push carlofinnegan/userservice:latest
sudo docker push carlofinnegan/borrowservice:latest
sudo docker push carlofinnegan/bookservice:latest

#Convert from compose
kompose convert -f docker-compose.yml

#Applied all Kubernetes configurations:
kubectl apply -f .

#Established port forwarding for each service done in separate terminals to ensure it was working
kubectl port-forward service/book 5006:5006
kubectl port-forward service/borrow 5008:5008
kubectl port-forward service/user 5002:5002
kubectl port-forward service/database 5432:5432
kubectl port-forward service/rabbitmq 5672:5672 15672:15672

Then I proceeded as seen in the video attached. It can be used to verify service is working etc